Security Analysis of Linearly Filtered NLFSRs

Mohammad Ali Orumiehchiha[1], Josef Pieprzyk[1], Ron Steinfeld[2] and Harry Bartlett [3]

[1]Center for Advanced Computing, Algorithms and Cryptography, Department of Computing,
Faculty of Science, Macquarie University, Sydney, NSW 2109, Australia
{mohammad.orumiehchiha,josef.pieprzyk}@mq.edu.au
[2]Clayton School of Information Technology
Monash University, Clayton VIC 3800, Australia
ron.steinfeld@monash.edu
[3]Information Security Institute, Queensland University of Technology,
126 Margaret Street, Brisbane Qld 4001, Australia
h.bartlett@qut.edu.au

Abstract. Our contributions are applying distinguishing attack on Linearly Filtered NLFSR as a primitive or associated with filter generators. We extend the attack on linear combinations of Linearly Filtered NLFSRs as well. Generally, these structures can be examined by the proposed techniques and the criteria will be achieved to design secure primitive. The attacks allow attacker to mount linear attack to distinguish the output of the cipher and recover its internal state. Also, we investigate security of the modified version of Grain stream cipher to present how invulnerable is the scheme against distinguishing attacks.

Keywords: Non-linear feedback shift register, Linearly Filtered NLFSR, Cryptanalysis, Key Recovery Attack, Distinguishing Attack.

1 Introduction

The one-time pad is the only cipher that is unbreakable even for an adversary who has unlimited computational power. Instead of a truly random sequence of bits, stream ciphers produce a pseudorandom sequence from a relatively short random sequence (also called the seed). This, however, has a profound impact on their security. Stream ciphers do not inherit the unconditional security - their security is conditional and depends on how difficult the adversary can recover the seed from an observed keystream.

The main advantage of stream ciphers is that they can be implemented very efficiently both in software and hardware making them very popular in the telecommunication industry. They are extensively used in the mobile communication providing the basic security tool to ensure

confidentiality and integrity of communication. Historically, first stream ciphers were built using shift registers with a linear feedback. Linear feedback shift registers (LFSR) modify their internal state by using a linear recursion. It turns out that LFSR with no nonlinear components are insecure and easy to break.

There are few distinct methods to design stream ciphers using LFSRs and some non-linear components. The design methods have been analysed thoroughly. Consequently, a collection of design criteria has been identified. The collection can be used by the designers to create new stream ciphers whose security can be tested using the developed cryptographic attacks. The most effective tests for stream cipher include the correlation and fast correlation attacks [22, 13, 25, 7] and the algebraic and fast algebraic attacks [6, 8, 1, 16].

A natural evolution in the design of stream ciphers was the introduction of non-linear feedback shift registers (NLFSRs). NLFSRs can be seen as a generalisation of LFSRs, where the modification of the internal state is done using a nonlinear relation [15]. While the mathematics behind LFSRs is well understood, the theory of NLFSRs is in its infancy stage. There are many basic problems related to NLFSRs still open. For instance, we do not know how to determine the period, identify different cycles, or find out the linear complexity of NLFSRs.

The lack of understanding of mathematics behind NLFSRs has led to proliferation of stream cipher designs based on NLFSRs. The finalist of the e-Stream project includes the Trivium [5] and Grain [17] ciphers that are exploiting one or several NFSRs combined with LFSRs. The security of a NLFSR filtered by a linear boolean function has been investigated against algebraic and correlation attacks in [2, 11]. In particular, the authors of [2] show that a linearly filtered non-linear feedback shift register (LF-NFSR) can be translated to a well-known *filter generator* including a LFSR and a non-linear boolean function as the filter function. Figure 1 illustrates the main contribution of the work presented in [2].

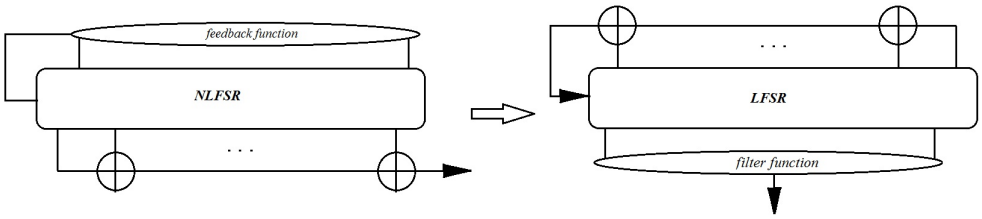

Fig. 1. LF-NLFSR can be considered as non-linear filter generator

1.1 Our Contribution

The paper investigates the design principles and security level of steam ciphers that are built from a LF-NLFSR. First, we introduce a taxonomy for generation of sequences obtained from stream ciphers that are built using a LF-NFSR. Next, we examine the security of these schemes against distinguishing attacks. Then, we identify criteria that need to be satisfied for a secure LF-NLFSRs. Finally, based on the proposed criteria, we show how to improve the time and data complexity of algebraic attacks on LF-NLFSR presented in [2].

This paper is organised as follows. Section 2 describes of the LF-NLFSR cipher and introduces the main idea behind our distinguishing attack. Section 3 investigates security properties of the stream ciphers whose LF-NLFSR are chosen at random. The security properties of LF-NLFSRs associated with NFSRs are studied in Section 4. In section 5, we study security of a stream cipher, which is based on linear combination of LF-NLFSRs. We prove that this type of cipher may be vulnerable to distinguishing attacks. In section 6, we suggest the design criteria to design stream ciphers based on LF-NLFSRs. Finally, Section 7 concludes the paper.

2 Description of LF-NLFSR

Pseudo-random sequences generated by stream ciphers based on LFSRs have been exhaustively studied and there is a good understanding of their statistical and cryptographic properties. To make the sequences immune against algebraic attacks, the (linear) sequence generated by a LFSR is filtered by a non-linear boolean function. The stream ciphers based on LFSRs and non-linear filters have attracted a lot of attention resulting in a large number of publications. For instance, works [23, 3, 20] present designs of stream ciphers using non-linear filters of linear sequences. Their security is analysed in [14, 26, 24].

The duality between stream ciphers based on non-linear filters of LFSR sequences and stream ciphers built from LF-NLFSRs is investigated in [2, 11]. The main idea is to replace a LFSR with non-linear filter by an appropriate NLFSR whose output sequence is filtered by a simple linear function. Thus, to determine the equivalent LF-NLFSR, one needs to define an update function as non-linear feedback function that is necessary to construct the NLFSR. Formally, the LF-NLFSR can be considered

as one $n-$bit NLFSR and a linear function L defined as follows:

$$\begin{aligned} s^t[i] &= s^{t-1}[i+1] && 0 \leq i < n-1 \\ s^t[i] &= f(s^{t-1}[0], s^{t-1}[1], \cdots, s^{t-1}[n-1]) && i = n-1 \end{aligned}$$

where $s^t[i]$ is i-th bit of the internal state of the NLFSR at time t. The output keystream is generated as follows:

$$z^t = L(s^{t-1}[0], s^{t-1}[1], \cdots, s^{t-1}[n-1])$$

In [2], this structure has been investigated in terms of algebraic and correlation attacks.

2.1 Attacks on LF-NLFSR

The LF-NLFSR can be vulnerable to distinguishing and state recovery attacks. Also, the attacks can be more efficient if the linear filter function has been chosen randomly. This section proposes a distinguishing attack scenario against a stream ciphers built on LF-NLFSR. The attack exploits linear relations between output bits and internal state of the NLFSR. It approximates the non-linear feedback function by the nearest affine function and thus establishes probabilistic linear relations. By using these probabilistic linear relations, the adversary can also recover internal state of the LF-NLFSR. The attack works even when the NLFSR uses a highly non-linear feedback function. The difference between our proposed attack and the attack from [2] is that the distinguishing attack only needs to approximate a small number of bits of the non-linear functions. This leads the adversary to find a distinguisher with high probability.

2.2 Distinguishing attack on LF-NLFSR

In this section, we show how to apply distinguishing attacks on stream ciphers based on LF-NLFSR (see Figure 2). To make the presentation clearer for the reader, we start from a simple example shown below.

Example 1: Given a $7-$bit NLFSR that generates output sequences by using the linear boolean function $L(s_1, s_3, s_4, s_7) = s_1 \oplus s_3 \oplus s_4 \oplus s_7$, where $s_i \quad (i = 1, \cdots, 7)$ is the i-th bit of the initial state of the NLFSR. The feedback function is a balanced non-linear boolean function $f(s_1, s_2, s_3, s_5, s_6, s_7) = s_1 \oplus s_2 \oplus s_6 \oplus (s_3 \cdot s_5 \cdot s_7)$. The NLFSR provides non-linear sequences with period $T_7 = 2^7 - 1$ [10] (see Figure 2). The output bits can be generated as follows:

$$O_i = s_{i+1} \oplus s_{i+3} \oplus s_{i+4} \oplus s_{i+7} \tag{1}$$

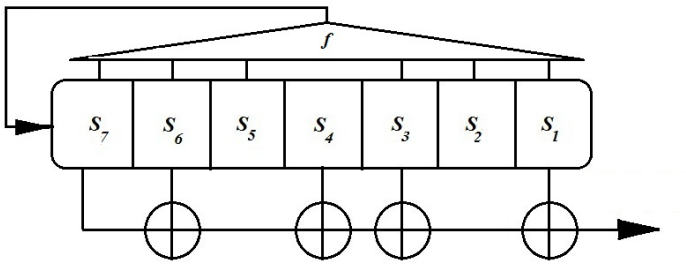

Fig. 2. a 7−bit LF-NLFSR as toy cipher

Now, the adversary can replace new generated bits in the internal state by a linear combination of initial state and output bits. In our example, we can rewrite s_{i+7} $(i \geq 0)$ as follows:

$$\begin{cases} s_7 &= s_1 \oplus s_3 \oplus s_4 \oplus O_1 \\ s_8 &= s_5 \oplus s_4 \oplus s_2 \oplus O_2 \\ s_9 &= s_6 \oplus s_5 \oplus s_3 \oplus O_3 \\ s_{10} &= s_1 \oplus s_3 \oplus s_6 \oplus O_1 \oplus O_4 \\ s_{11} &= s_2 \oplus s_1 \oplus s_3 \oplus O_1 \oplus O_2 \oplus O_5 \\ s_{12} &= s_3 \oplus O_3 \oplus s_4 \oplus s_2 \oplus O_2 \oplus O_6 \\ s_{13} &= s_3 \oplus O_4 \oplus s_5 \oplus O_3 \oplus s_4 \oplus O_7 \\ s_{14} &= O_5 \oplus s_6 \oplus O_4 \oplus s_5 \oplus s_4 \oplus O_8 \\ s_{15} &= s_3 \oplus s_4 \oplus O_6 \oplus s_1 \oplus O_1 \oplus O_5 \oplus s_6 \oplus s_5 \oplus O_9 \\ s_{16} &= s_3 \oplus s_5 \oplus O_7 \oplus s_2 \oplus O_2 \oplus O_6 \oplus s_1 \oplus O_1 \oplus s_6 \oplus O_{10} \\ s_{17} &= s_6 \oplus O_8 \oplus O_3 \oplus O_7 \oplus s_2 \oplus O_2 \oplus s_1 \oplus O_1 \oplus O_{11} \\ s_{18} &= s_4 \oplus s_1 \oplus O_1 \oplus O_9 \oplus O_4 \oplus O_8 \oplus O_3 \oplus s_2 \oplus O_2 \oplus O_{12} \\ s_{19} &= s_2 \oplus s_3 \oplus s_5 \oplus O_2 \oplus O_3 \oplus O_4 \oplus O_5 \oplus O_9 \oplus O_{10} \oplus O_{13} \\ s_{20} &= s_3 \oplus s_4 \oplus s_6 \oplus O_3 \oplus O_4 \oplus O_5 \oplus O_6 \oplus O_{10} \oplus O_{11} \oplus O_{14} \\ s_{21} &= s_1 \oplus s_3 \oplus s_5 \oplus O_1 \oplus O_4 \oplus O_5 \oplus O_6 \oplus O_7 \oplus O_{11} \oplus O_{12} \oplus O_{15} \end{cases} \tag{2}$$

In addition to Equations (2), each new generated internal state bit can be approximated by a linear approximation of the feedback function of NLFSR. Also, we have:

$$Pr(f(s_1, s_2, s_3, s_5, s_6, s_7) = s_1 \oplus s_2 \oplus s_6) = 1 - 2^{-3} = \frac{1}{2} + \frac{3}{8} \tag{3}$$

By applying linear approximations for the generated bits in the internal state of NLFSR, the adversary can derive probabilistic linear relations, which are likely to be biased. For instance, the adversary can find a biased relation by xoring O_2, O_3 and O_{15} as shown below

$$\begin{cases} O_2 & = s_5 \oplus s_4 \oplus s_1 \oplus s_6 \\ O_3 & = s_6 \oplus s_5 \oplus s_3 \oplus s_2 \oplus s_1 \oplus s_4 \oplus O_1 \\ O_{15} & = s_2 \oplus s_3 \oplus O_2 \oplus O_3 \oplus O_4 \oplus O_7 \oplus O_8 \oplus O_{10} \oplus O_{11} \oplus O_{12} \oplus O_{13}. \end{cases} \tag{4}$$

We can rewrite the following probabilistic linear relation:

$$O_1 \oplus O_4 \oplus O_7 \oplus O_8 \oplus O_{10} \oplus O_{11} \oplus O_{12} \oplus O_{13} \oplus O_{15} = 0 \tag{5}$$

We know that each relation of Equation 4 holds with the probability $1 - 2^{-3}$. Therefore, we have:

$$Pr(O_1 \oplus O_4 \oplus O_7 \oplus O_8 \oplus O_{10} \oplus O_{11} \oplus O_{12} \oplus O_{13} \oplus O_{15} = 0) = \frac{1}{2} + (2^2 \cdot (\frac{3}{8})^3) = \frac{1}{2} + 2^{-2.245}. \tag{6}$$

Example 1 uses three linear approximations and establishes a linear distinguisher based on the output keystream bits. One would ask if there is an upper bound on the number of linear approximations for the non-linear function. Theorem 1 gives such an upper bound.

Theorem 1. *Let LF-NLFSR N be an n-bit NLFSR with feedback function f and linear filter function L. If the best linear approximation of f is ℓ such that*

$$Pr(f = \ell) = \frac{1}{2} + \epsilon_f$$

Then, having $n + 1$ consecutive bits of the keystream outputs, there is at least one biased linear function.

Proof. The proof can be derived from [12]. □

The smallest number of output bits required to find a biased linear function (ℓ_p) depends on the linear filter function ℓ and the feedback function f. In general, if all $n+1$ output bits are involved in ℓ_p (*e.g.* $n+1$ linear approximations), then

$$Pr(\ell_p = 0) = \frac{1}{2} + 2^n \cdot \epsilon_f^{(n+1)}$$

Note that Theorem 1 shows that the security of the cipher cannot be better than $\epsilon_f^{-2\cdot(n+1)}$. For each relation, we need to use at least one linear approximation with the probability $P_L = 1/2 + \epsilon$. Assume that with m linear equation, the adversary could find a biased relation for the output keystream bits with the probability $P = 1/2 + (2^{m-1} \cdot \epsilon^m)$, then the attack will be successful if

$$P < 2^{\mathbb{k}/2},$$

where $\mathbb{k}$ is the secret key space of the cipher. In other words, the bias in the relation will be $\epsilon' = 2^{m-1} \cdot \epsilon^m$ and hence the attack will be faster than the $O(2^{\mathbb{k}})$ run-time of exhaustive search if $(\epsilon')^{-2} < 2^{\mathbb{k}/2}$.

There is a trend in the design of cryptographic components and systems, in which they are chosen at random. The main justification for this is the belief that random choice can prevent the cryptographic system against new yet unknown attacks. In the next section, we analyse the stream cipher based on LF-NLFSR when both the linear filter function and the non-linear feedback function are chosen at random.

3 Random LF-NLFSR

A random LF-NLFSR is a LF-NLFSR whose linear filter function and feedback function have been generated randomly. More precisely, the non-linear feedback function is chosen at random from all balanced non-linear functions. The linear filter function is chosen randomly and uniformly from the set of all linear functions (excluding the constants).

3.1 Cryptanalysis of Random LF-NLFSRs

To analyse the security of a random LF-NLFSR, we need the following two theorems. The first theorem evaluates the probability that a set of p randomly chosen q-tuples over a finite field $\mathbb{F}_2$ consists of linear independent tuples (over a $\mathbb{F}_2$). We take advantage of [19] that provides the following statement.

Theorem 2. *([19]) Let $M_{q,q+p}$ be a $q \times (q+p)$ random matrix, over the finite field $\mathbb{F}_2$ where $-q \leq p \leq 0$. If $\rho(M)$ is rank of matrix M, then we have,*

$$P(\rho(M_{q,q+p}) = q+p)) = \prod_{j=0}^{q+p-1} (1 - \frac{1}{2^{q-j}}), \quad -q \leq p \leq 0.$$

Proof. Proof can be found in [19]. □

In general, the probability that a random $q \times (q+p)$ binary matrix $M_{q,q+p}$ is of the full rank q for $p \geq 0$, for large q is:

$$P(\rho(M_{q,q+p}) = q) = \prod_{i=p+1}^{\infty} (1 - \frac{1}{2^i}), \quad p = 0, 1, \cdots .$$

An interesting observation proved in [4] is that for a matrix defined as in Theorem (2), on the average, one would need two extra columns only to achieve the full rank. This result does not depend on q. For 7 or 8 extra columns, the probability of achieving the full rank is very close to 1.

Theorem 3. *Let matrix $M_{q,q+p}$,$-q \leq p \leq 0$, is a random binary matrix that the matrix entries are chosen independently and uniformly, then probability that the rank of matrix M equals lesser than $q + p$ is:*

$$P(\rho(M_{q,q+p}) < q+p) = 1-P(\rho(M_{q,q+p}) = q+p)) = 1- \prod_{j=0}^{q+p-1} (1-\frac{1}{2^{q-j}}), \quad -q \leq p \leq 0.$$

Proof. The rank matrix M is up to $min(q, p+q) = p+q$. Therefore, probability that the rank of matrix M is lesser than $q+p$ is $1-P(\rho(M_{q,q+p}) = q + p))$. Based on Theorem (2), the probability will be $1 - \prod_{j=0}^{q+p-1}(1 - \frac{1}{2^{q-j}})$, where $-q \leq p \leq 0$. □

By using Theorems (2,3), one can find out the lower bound of security in random LF-NLFSRs given below.

Theorem 4. *The number of observed keystream bits (N_m) to find at lease one linear biased relation (distinguisher) using m linear approximations should satisfy*

$$\pi(n, m)^{-1} = \binom{N_m}{m},$$

where $\pi(n, m)$ is the probability of finding at least one linear dependency for the corresponding matrix of a n-bit Random LF-NLSR.

Proof. By using Theorem 3, the probability of finding at least one linear dependency for the corresponding matrix of a n-bit random LF-NLFSR can be computed as

$$\pi(n, m) = 1 - \prod_{j=0}^{n-m-1} (1 - \frac{1}{2^{n-j}}),$$

where m is the number of the row. So, the number of $m \times n$ matrices which should be checked to find at least one linear dependency with probability near to one is $\frac{1}{\pi(n,m)}$. The adversary needs to check all combinations of m linear equations from the required keystream bits (N_m), *e.g.*

$$\pi(n,m)^{-1} = \binom{N_m}{m}$$

□

Theorem 4 shows for 64-bit random LF-NLFSR, the probability of finding a linear biased relation by applying linear approximation for two and four output bits is 2^{-64} and $2^{-61.19}$, respectively. The required keystream bits to apply the attack is $2^{32.48}$ and $2^{21.25}$, respectively.

We may expect the matrices might look random even if the feedback/filter function is not chosen at random in many cases, so the attack can apply even for schemes with non-random feedback/filter function. Note that we consider balanced non-linear functions and our assumptions do not limit us to a certain class of Boolean functions. If the adversary finds a linear biased relation using m linear approximations, then he just needs to approximate feedback function m times and the probability of linear distinguisher can computed as follows:

$$Pr(\textit{linear distinguisher}) = 1/2 + 2^{m-1} \cdot (\epsilon_f^m).$$

Therefore, the data complexity of the attack to distinguish output of LF-NLFSR from a truly random binary source will be around $O(\epsilon_f^{-2 \cdot m})$.

As the result, to apply a distinguishing attack on a random LF-NLFSR two main phases are needed; pre-processing and on-line phases. In the pre-processing phase, the adversary tries to find distinguisher (or distinguishers). Theorem 4 determines the probability of finding a valid distinguisher and the required data complexity in the pre-processing phase. After finding the distinguisher, a distinguishing attack will be applied in the on-line phase.

4 LF-NLFSR and LFSR

Another extension in producing non-linear sequence generated by LF-NLFSR is combining the output of LF-NLFSR with a non-linearly filtered linear sequences. The Grain stream cipher [17, 18] uses this structure to produce keystream output bits. Some security analyses have been published in [2, 21, 9]. Figure 3 shows the structure of the design, which combines a LF-NLFSR with a filter generator.

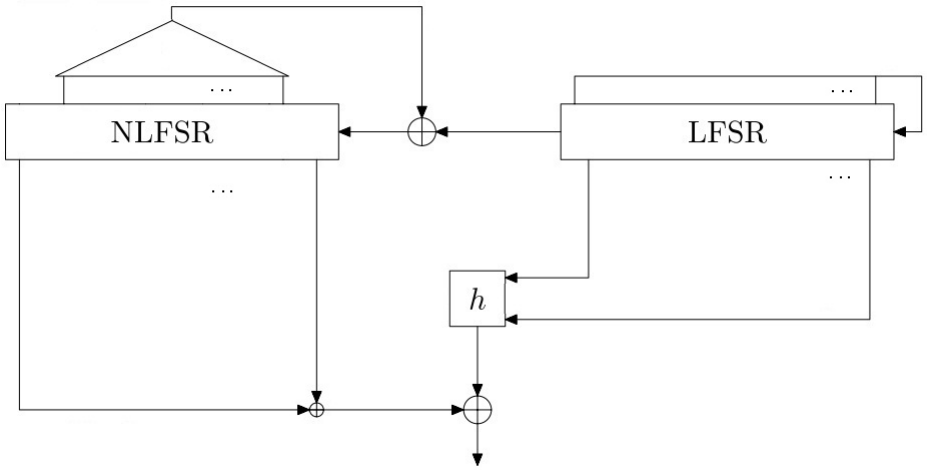

Fig. 3. LF-NLFSR combining with a filter generator

4.1 Applying the attack on modified version of Grain [2].

In this case, we are dealing with the following equations:

$$x_t = \bigoplus_{i\in\alpha} z_i \oplus \bigoplus_{j\in\beta} x_j \oplus \bigoplus_{k\in\gamma} y_k \oplus h^t(y_0, \cdots, y_m)$$

where x_i and y_i are $i-$bit of the internal states of NLFSR and LFSR, respectively and α, β, and γ are sets certifying, which combination of output bits, NLFSR and LFSR states are effective to make new NLFSR bit. $h^t(y_0, \cdots, y_m)$ is the output of filter function h after t clocks. To apply distinguishing attack, one first should approximate h function by one or more linear boolean function and then find biased relation as mentioned before. For every non-linear Boolean function, one may find one or more affine functions, which are near to the function. Some of them are the nearest linear Boolean functions. In general, the adversary needs to have more linear relations to increase the probability of finding distinguisher.

Since in every output bit, exactly two non-linear functions (non-linear feedback function and h function) have been involved, if the adversary can find a linear biased relation by xoring two output keystream bits, then the number of approximations will be four; two approximations for new generated bits and other two approximations are related h function. Consequently, we can have:

$$Pr(z_x \oplus z_y = 0) = \frac{1}{2} + 2^3 \cdot (\epsilon_f^{-2} \cdot \epsilon_h^{-2}),$$

where ϵ_f and ϵ_h indicate the biases of linear approximation of non-linear feedback function f and non-linear filter h, respectively.

5 Linear Combination of LF-NLFSRs

A different method to design non-linear sequences by using LF-NLFSRs is a linear combination of two or more LF-NLFSRs. In fact, if the cipher is a linear combination of several LF-NLFSRs then we call it LC-NLFSR in the rest of the paper. Assume that $O_1^t, \cdots, O_m^t$ are output sequences of m distinct LF-NLFSR at time t. Then the output of the cipher (O^t) can be produced as follows:

$$O^t = O_1^t \oplus O_2^t \oplus \cdots \oplus O_m^t$$

LC-NLFSR structure is illustrated in Figure 4. Although, the attacks from [2] cannot be applied to LC-NLFSR, we will show that a LC-NLFSR is vulnerable to distinguishing attacks.

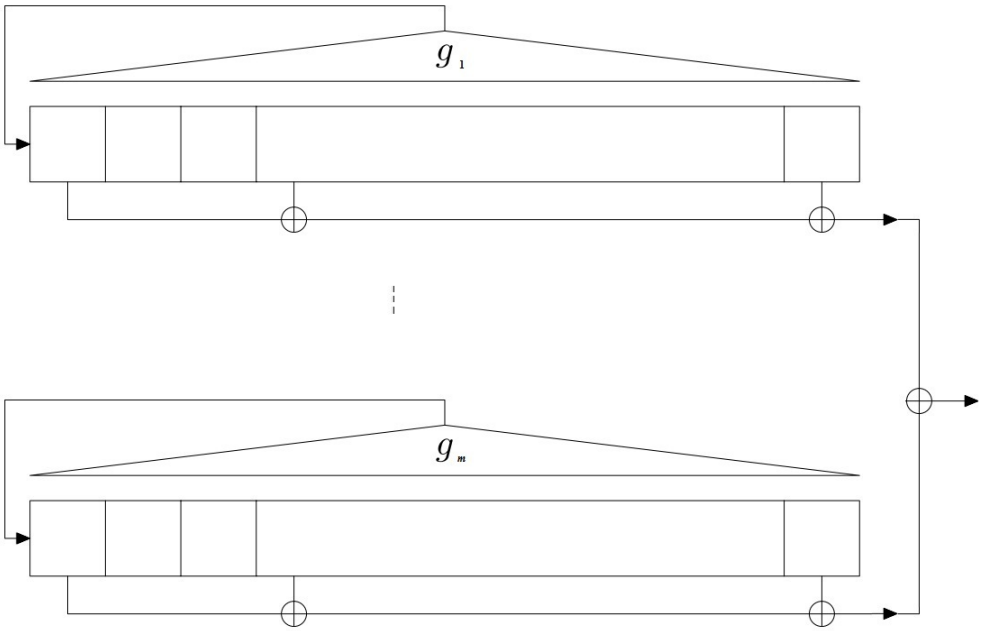

Fig. 4. Linear Combination of LF-NLFSRs (LC-NLFSR)

5.1 Distinguishing Attack on LC-NLFSRs

The previous section deals with security analysis of a single LF-NLFSR and its resistance against the distinguishing attack. At ESC 2008, C. Berbain presented the results of his work [2] and mentioned few open problems. One of them was analysis of a linear combination of two LF-NLFSR. In this section, we investigate security of linearly combined two LF-NLFSRs (LC-NLFSRs). We present an analysis and criteria to design LC-NLFSR schemes.

Example 2: Let N_1 and N_2 be two LF-NLFSRs (with non-linear feedback function g_1 and g_2 and linear filter functions L_1 and L_2), which have been linearly combined together to generate keystream bits (O_t at time $t \geq 0$). Let P_1 and P_2 be linear combinations of internal states of N_1 and N_2 respectively (See Figure 5). We obviously know that:

$$P_1^t \oplus P_2^t = O_t,$$

where P_i^t is a linear filter of state shift register N_i at time t and $i \in \{1, 2\}$.

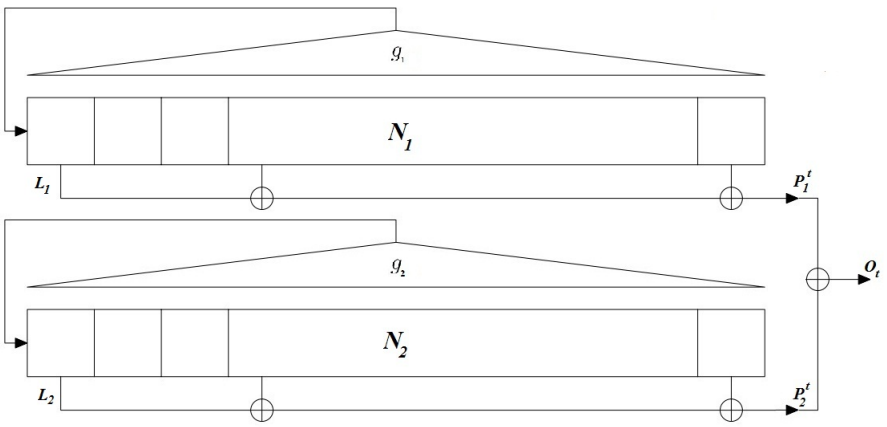

Fig. 5. LC-NLFSR of Example 2

Based on the method proposed in Section 2.1, we assume that the adversary has found two different biased linear relations $\lambda = \bigoplus_{i \in \{\phi_1\}} P_1^i$ and $\mu = \bigoplus_{i \in \{\phi_2\}} P_2^i$ for each NLFSR N_1 and N_2, where ϕ_1 and ϕ_2 represent all effective coefficients to build linear biased relations. Clearly, the adversary cannot use the biased relations λ and μ to find a linear bias of the output bits, because the sets ϕ_1 and ϕ_2 are not necessarily the

same. It means that we need to find linear biased relations derived from two LF-NLFSRs in the same instance to exploit these relations to find a linear biased relation based on output keystream bits. To do this, we can consider linear biased relations λ and μ in the following polynomial forms:

$$\lambda(x) = c_0 + c_1 x + c_2 x^2 + \cdots + x^{l1}$$

$$\mu(x) = d_0 + d_1 x + d_2 x^2 + \cdots + x^{l2}$$

where $c_i, d_i \in F_2$ are coefficients polynomials $\lambda(x)$ and $\mu(x)$ and the polynomials degrees are $l_1,\ l_2$, respectively, where $l_1 > N_1,\ l_2 > N_2$.

To find a linear biased relation, which is valid for the output keystream bits, we can multiply $\lambda(x)$ and $\mu(x)$. In this case, the number of involved relations in multiplied polynomial will be higher than the relations involved in each polynomial $\lambda(x)$ and $\mu(x)$. So, it is more efficient if we could find the polynomial with lower possible relations.

A different approach is to finding the lowest degree polynomial $\Lambda(x)$ satisfying following conditions:

1. $\lambda(x)|\Lambda(x)$
2. $\mu(x)|\Lambda(x)$

where $f(x)|g(x)$ means *g(x)* divides *f(x)*. Note that in addition to LF-NLFSR and LC-NLFSR, the proposed distinguishing attack can be successfully applied to the case m LF-NLFSRs are linearly combined with n filter generators. For $m = 1,\ n = 1$, the authors of [2] have investigated the security of the cipher against algebraic and correlation attacks, but the attacks are not applicable for the cases $m > 1$ and $n > 1$.

6 Linear Filtering Properties

The interesting question is whether there is any condition to design a linear filter function in LF-NLFSR. To answer this question, the following theorem can be helpful. First we need to define some concepts. Let g be a monic polynomial over F_q. We call g a *characteristic polynomial* of σ if the linear operator $g(T)$ annihilates σ, i.e. $g(T)\sigma = 0$, where 0 denotes the zero sequence of V (the sequence all of whose terms are 0). For any periodic sequence $\sigma \in V$,

$$J_\sigma = g \in F_q[x]\ :\ g(T)\sigma = 0$$

is a non-zero ideal (called the T -annihilator of σ) in the principal ideal domain $F_q[x]$. The uniquely determined monic polynomial $m_\sigma \in F_q[x]$ with $J_\sigma = (m_\sigma) = m_\sigma F_q[x]$ is called the minimal polynomial of σ. Thus the characteristic polynomials of σ are precisely the monic polynomials in $F_q[x]$ that are multiples of m_σ. Note that the degree of m_σ is called the linear complexity $L(\sigma)$ of σ. In [11], a method has been explained to compute the minimal polynomial of a periodic sequence from a known characteristic polynomial and a suitable number of initial terms of the sequence.

Theorem 5. *[11] Let $A = (a_i)_{i=0}^{\infty}$ be a periodic binary sequence with minimal polynomial $p_a \in F_2[x]$ and let $L_\alpha = \alpha_1 + \alpha_2 x + \cdots + \alpha_n x^{n-1}$ be a non-zero polynomial over F_2. Then, the sequence*

$$B = (b_i)_{i=0}^{\infty} = (\alpha_1 a_{i+n} + \alpha_2 a_{i+n-1} + \cdots + \alpha_n a_i)_{i=0}^{\infty}$$

is periodic and its minimal polynomial is given by $p_b = \frac{p_a}{gcd(p_a, L_\alpha)}$

Note: These investigations allow designers to draw new rules for designing stream ciphers based on LF-NLFSR. Let $A = (a_i)_{i=0}^{T}$ be non-linear sequence generated by a NLFSR, with minimal polynomial $p_a \in F_2[x]$. To design a linear filter L_α achieving maximum period of sequence A, p_a and L_α) should be co-prime. When p_a has been divided by L_α, the output period will not achieve the period of NLFSR. From this point, the importance of designing full period NLFSR has been indicated. Because even if NLFSR generates several long period sequences, but the linearly filtered output sequences may have shorter period. Either NLFSR does not have a long full period or generates a long non-linear cycle, therefore the best choice to design linear filter function is an irreducible polynomial. Theorem 6 describes a criterion to design the linear filter function.

Theorem 6. *[11] Let A be periodic binary sequence generated by a $n-$bit NLFSR with period 2^{n-1} (all nonzero n-bit states). The output sequences B have the same period and linear complexity if the canonical factorization of the filter polynomial contains only irreducible factors equal to x or $x-1$, or whose degrees do not divide n.*

6.1 Some observations on Grain based LF-NLFSR [2]

The LF-NLFSR proposed in [2], has been taken the NLFSR from Grain version 1.0 [17] and outputs keystream bits by applying a linear filter function on internal state of the NLFSR.

The 80-bit NLFSR has the feedback function f given as follows:

$$\begin{aligned}
s_{t+80} =& f(s_t, s_{t+1}, \cdots, s_{t+79}) \\
=& s_{t+62} \oplus s_{t+60} \oplus s_{t+52} \oplus s_{t+45} \oplus s_{t+37} \oplus s_{t+33} \oplus s_{t+28} \oplus s_{t+21} \\
& \oplus s_{t+14} \oplus s_{t+9} \oplus s_t \oplus s_{t+63}s_{t+60} \oplus s_{t+37}s_{t+33} \oplus s_{t+15}s_{t+9} \\
& \oplus s_{t+60}s_{t+52}s_{t+45} \oplus s_{t+33}s_{t+28}s_{t+21} \oplus s_{t+63}s_{t+45}s_{t+28}s_{t+9} \\
& \oplus s_{t+60}s_{t+52}s_{t+37}s_{t+33} \oplus s_{t+63}s_{t+60}s_{t+21}s_{t+15} \oplus s_{t+63}s_{t+60}s_{t+52}s_{t+45}s_{t+37} \\
& \oplus s_{t+33}s_{t+28}s_{t+21}s_{t+15}s_{t+9} \oplus s_{t+52}s_{t+45}s_{t+37}s_{t+33}s_{t+28}s_{t+21}
\end{aligned}$$

The keystream bits can be generated by the following linear function:

$$O_t = s_{t+1} \oplus s_{t+2} \oplus s_{t+4} \oplus s_{t+10} \oplus s_{t+31} \oplus s_{t+43} \oplus s_{t+56} \oplus s_{t+63}$$

If the linear filter function is not designed properly, then the proposed attacks [2] can be applied more efficiently. As mentioned in [2], the size of the blocks of equations of constant degree is determined by the difference between the position of the highest tap index in the update function and the position updated by the feedback function. It means that $(80-63) = 17$ bits of internal state can be written by linear combination of other internal state bits. It decreases the number of independent variables from 80 bits to 63 bits. The algebraic technique, proposed in [2], keeps the degree of the corresponding system fixed and applies an algebraic attack to recover the internal state of the NLFSR.

$$\begin{aligned}
s_{80} &= O_{17} \oplus s_{76} \oplus s_{60} \oplus s_{48} \oplus s_{27} \oplus s_{21} \oplus s_{19} \oplus s_{18} \\
s_{81} &= O_{19} \oplus s_{78} \oplus s_{62} \oplus s_{50} \oplus s_{29} \oplus s_{23} \oplus s_{21} \oplus s_{20} \\
s_{83} &= O_{20} \oplus s_{79} \oplus s_{63} \oplus s_{51} \oplus s_{30} \oplus s_{24} \oplus s_{22} \oplus s_{21} \\
s_{84} &= O_{21} \oplus s_{80} \oplus s_{64} \oplus s_{52} \oplus s_{31} \oplus s_{25} \oplus s_{23} \oplus s_{22}
\end{aligned} \tag{7}$$

The important point which has not been investigated in [2] is the critical role of the linear filter function in the security of the cipher. Now, we mention some observations about the effects of the linear filter function on security of the LF-NLFSR.

Lemma 1. *The number of the independent variables in System 7 is* 63.

Proof. All new internal state bits (s_{t+80}, $t \geq 0$) generated by the update function can be written by $(s_{17}, \cdots, s_{79})$ variables. In other words, the number of the independent variables in System 7 is $80 - 17 = 63$.

□

Lemma 1 shows the complexity of solving non-linear system will be dramatically decreased.

Observation 1: Linear System 7 has been generated by a specific polynomial called generating polynomial. It is proved that the linear system inherits mathematical properties from the generating polynomial. If the polynomial is not primitive, then the linear equations will be repeated with period less than $2^{80-17} - 1$. Note that because of dependency of new generated variables on the variables $(s_{17}, \cdots, s_{79})$ and output bits $(O_t, \quad t \geq 0)$, the new variables will not exactly repeated but the linear combination of the independent variables are the same. Consequently, linear complexity of combination of the output bits will be surprisingly decreased. This property leads attacker to compute linear complexity of the keystream bits.

Assume the period of repetition of linear relations of $(s_{17}, \cdots, s_{79})$ is T, then O_t and O_{t+T} have the following relation:

$$O_t \oplus O_{t+T} = \bigoplus_{\tau=0}^{T} \alpha_\tau O_{t+\tau}$$

where $\alpha_\tau \in F_2$ depends on the linear filter function. Now, we illustrate the results with considering Example 3.

Example 3: In Example 1, the period of NLFSR state is $T_7 = 2^7 - 1$, but one can find the repetition of linear equations in the internal state in

a period less than T_7. For instance, we have:

$$
\begin{cases}
s_7 &= s_1 \oplus s_3 \oplus s_4 \oplus O_1 \\
s_8 &= s_5 \oplus s_4 \oplus s_2 \oplus O_2 \\
s_9 &= s_6 \oplus s_5 \oplus s_3 \oplus O_3 \\
s_{10} &= s_1 \oplus s_3 \oplus s_6 \oplus O_1 \oplus O_4 \\
s_{11} &= s_2 \oplus s_1 \oplus s_3 \oplus O_1 \oplus O_2 \oplus O_5 \\
s_{12} &= s_3 \oplus O_3 \oplus s_4 \oplus s_2 \oplus O_2 \oplus O_6 \\
s_{13} &= s_3 \oplus O_4 \oplus s_5 \oplus O_3 \oplus s_4 \oplus O_7 \\
&\dots \\
s_{38} &= s_1 \oplus s_3 \oplus s_4 \oplus O_1 \oplus O_7 \oplus O_9 \\
&\oplus O_{10} \oplus O_{11} \oplus O_{13} \oplus O_{14} \oplus O_{16} \oplus O_{18} \\
&\oplus O_{21} \oplus O_{22} \oplus O_{23} \oplus O_{24} \oplus O_{28} \oplus O_{29} \oplus O_{32} \\
s_{39} &= s_5 \oplus s_4 \oplus s_2 \oplus O_2 \oplus O_8 \oplus O_{10} \\
&\oplus O_{11} \oplus O_{12} \oplus O_{14} \oplus O_{15} \oplus O_{17} \oplus O_{19} \\
&\oplus O_{22} \oplus O_{23} \oplus O_{24} \oplus O_{25} \oplus O_{29} \oplus O_{30} \oplus O_{33} \\
s_{40} &= s_6 \oplus s_5 \oplus s_3 \oplus O_3 \oplus O_9 \oplus O_{11} \\
&\oplus O_{12} \oplus O_{13} \oplus O_{15} \oplus O_{16} \oplus O_{18} \oplus O_{20} \\
&\oplus O_{23} \oplus O_{24} \oplus O_{25} \oplus O_{26} \oplus O_{30} \oplus O_{31} \oplus O_{34} \\
s_{41} &= s_1 \oplus s_3 \oplus s_6 \oplus O_1 \oplus O_4 \oplus O_{10} \\
&\oplus O_{12} \oplus O_{13} \oplus O_{14} \oplus O_{16} \oplus O_{17} \oplus O_{19} \\
&\oplus O_{21} \oplus O_{24} \oplus O_{25} \oplus O_{26} \oplus O_{27} \oplus O_{31} \oplus O_{32} \oplus O_{35} \\
s_{42} &= s_2 \oplus s_1 \oplus s_3 \oplus O_1 \oplus O_2 \oplus O_5 \\
&\oplus O_{11} \oplus O_{13} \oplus O_{14} \oplus O_{15} \oplus O_{17} \oplus O_{18} \oplus O_{20} \\
&\oplus O_{22} \oplus O_{25} \oplus O_{26} \oplus O_{27} \oplus O_{28} \oplus O_{32} \oplus O_{33} \oplus O_{36} \\
s_{43} &= s_3 \oplus O_3 \oplus s_4 \oplus s_2 \oplus O_2 \oplus O_6 \\
&\oplus O_{12} \oplus O_{14} \oplus O_{15} \oplus O_{16} \oplus O_{18} \oplus O_{19} \oplus O_{21} \\
&\oplus O_{23} \oplus O_{26} \oplus O_{27} \oplus O_{28} \oplus O_{29} \oplus O_{33} \oplus O_{34} \oplus O_{37} \\
s_{44} &= s_3 \oplus O_4 \oplus s_5 \oplus O_3 \oplus s_4 \oplus O_7 \\
&\oplus O_{13} \oplus O_{15} \oplus O_{16} \oplus O_{17} \oplus O_{19} \oplus O_{20} \oplus O_{22} \\
&\oplus O_{24} \oplus O_{27} \oplus O_{28} \oplus O_{29} \oplus O_{30} \oplus O_{34} \oplus O_{35} \oplus O_{38}
\end{cases}
\tag{8}
$$

Relation 8 shows that the internal state of the NLFSR after just 31 clocks can be derived from previous states by adding a certain linear combinations of the output bits. More particularly, Relation 9 presents the relation between s_{38} and s_7.

$$s_{38} = s_7 \oplus O_7 \oplus O_9 \oplus O_{10} \oplus O_{11} \oplus O_{13} \oplus O_{14} \oplus O_{16} \oplus O_{18} \oplus O_{21} \oplus O_{22} \oplus O_{23} \oplus O_{24} \oplus O_{28} \oplus O_{29} \oplus O_{32} \quad (9)$$

In the case of Grain based LF-NLFSR, the polynomial derived by linear filter function is not irreducible and it can be divided to the following irreducible polynomial:

$$\begin{aligned} x^{80} + x^{76} + x^{60} + x^{48} + x^{27} + x^{21} + x^{19} + x^{18} = & (x+1)(x^3 + x + 1)(x^{18}) \\ & (x^7 + x^5 + x^4 + x^3 + 1) + \\ & (x^{14} + x^{13} + x^{11} + x^{10} + \\ & x^8 + x^6 + x^5 + x + 1) + \\ & (x^{37} + x^{35} + x^{34} + x^{32} + \\ & x^{30} + x^{25} + x^{24} + x^{23} + \\ & x^{21} + x^{17} + x^{16} + x^{10} + \\ & x^6 + x^5 + x^3 + x^2 + 1) \end{aligned}$$

Table 1 gives a comparison between the results presented in [2] and our new results on LF-NLFSR Grain Stream Cipher.

Table 1. Comparison between the results presented in [2] and our new results on LF-NLFSR Grain Stream Cipher

	Data Complexity	Time Complexity	The number of effective variables
[2]	2^{21}	2^{49}	80
Our Results	$2^{19.28}$	$2^{44.98}$	80-17=63

7 Conclusions

This work investigates security of stream ciphers based on LF-NLFSR. We firstly categorise key generations based on LF-NLFSR. We examine the security of LF-NLFSR, random LF-NLFSR and combination of LF-NLFSR and filter generators against distinguishing attack. Also, the security of the scheme being based on linear combination of LF-NLFSRs

is analysed. In addition, based on the proposed criteria, we presented an improvement on time and data complexity of algebraic attack on the Grain-like LF-NLFSR presented in [2].

References

1. F. Armknecht, *Improving fast algebraic attacks*, in FSE, 2004, pp. 65–82.
2. C. Berbain, H. Gilbert, and A. Joux, *Algebraic and correlation attacks against linearly filtered non linear feedback shift registers*, 5381 (2009), pp. 184–198.
3. A. Braeken, J. Lano, N. Mentens, B. Preneel, and I. Verbauwhede, *Sfinks: A synchronous stream cipher for restricted hardware environments*, in SKEW - Symmetric Key Encryption Workshop, 2005.
4. R. Brent, S. Gao, and A. Lauder, *Random krylov spaces over finite fields*, SIAM J. Discrete Math., 16 (2003), p. 276287.
5. C. D. Cannière and B. Preneel, *Trivium*, in The eSTREAM Finalists, 2008, pp. 244–266.
6. N. Courtois and W. Meier, *Algebraic attacks on stream ciphers with linear feedback*, in Advances in Cryptology - EUROCRYPT 2003,Warsaw, Poland, 2003, Proceedings, Springer, 2003, pp. 345–359.
7. N. T. Courtois, *Higher order correlation attacks, xl algorithm and cryptanalysis of toyocrypt*, in ICISC 2002, Springer-Verlag, 2002, pp. 182–199.
8. N. T. Courtois and W. Meier, *Fast algebraic attacks on stream ciphers with linear feedback*, in Crypto 2003, LNCS 2729, Springer, pp. 177–194.
9. I. Dinur and A. Shamir, *Breaking grain-128 with dynamic cube attacks*, in Proceedings of the 18th international conference on Fast software encryption, FSE'11, Berlin, Heidelberg, 2011, Springer-Verlag, pp. 167–187.
10. E. Dubrova, *A list of maximum period nlfsrs.*, IACR Cryptology ePrint Archive, 2012 (2012), p. 166.
11. B. M. Gammel and R. Göttfert, *Linear filtering of nonlinear shift-register sequences*, in WCC, 2005, pp. 354–370.
12. J. Goli, *Intrinsic statistical weakness of keystream generators*, in Advances in Cryptology ASIACRYPT'94, J. Pieprzyk and R. Safavi-Naini, eds., vol. 917 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 1995, pp. 91–103.
13. J. D. Golic, *Correlation via linear sequential circuit approximation of combiners with memory*, in EUROCRYPT, 1992, pp. 113–123.
14. J. D. Golic, A. Clark, and E. Dawson, *Generalized inversion attack on nonlinear filter generators*, IEEE Trans. Comput., 49 (2000), pp. 1100–1109.
15. S. W. Golomb, *Shift register sequences.*, Aegean Park Press, 1982.
16. P. Hawkes and G. G. Rose, *Rewriting variables: The complexity of fast algebraic attacks on stream ciphers*, in CRYPTO, 2004, pp. 390–406.
17. M. Hell, T. Johansson, and W. Meier, *Grain - a stream cipher for constrained environments*, ECRYPT Stream Cipher Project.
18. M. Hell, T. Johansson, and W. Meier, *Grain: a stream cipher for constrained environments*, IJWMC, 2 (2007), pp. 86–93.
19. V. F. Kolchin, *Random graphs*, Cambridge University Press, New York, NY, USA, 1999.
20. Y. Luo, Q. Chai, G. Gong, and X. Lai, *A lightweight stream cipher wg-7 for rfid encryption and authentication*, in GLOBECOM, 2010, pp. 1–6.

21. A. Maximov, *Cryptanalysis of the "grain" family of stream ciphers*, in Proceedings of the 2006 ACM Symposium on Information, computer and communications security, ASIACCS '06, ACM, 2006, pp. 283–288.
22. W. Meier and O. Staffelbach, *Fast correlation attacks on certain stream ciphers*, J. Cryptology, 1 (1989), pp. 159–176.
23. Y. Nawaz and G. Gong, *Wg: A family of stream ciphers with designed randomness properties*, Inf. Sci., 178 (2008), pp. 1903–1916.
24. M. A. Orumiehchiha, J. Pieprzyk, and R. Steinfeld, *Cryptanalysis of wg-7: a lightweight stream cipher*, Cryptography and Communications, (2012), pp. 1–9.
25. W. T. Penzhorn, *Correlation attacks on stream ciphers: Computing low-weight parity checks based on error-correcting codes*, in FSE, 1996, pp. 159–172.
26. S. Rønjom, G. Gong, and T. Helleseth, *A survey of recent attacks on the filter generator*, in Proceedings of the 17th international conference on Applied algebra, algebraic algorithms and error-correcting codes, AAECC'07, Berlin, Heidelberg, 2007, Springer-Verlag, pp. 7–17.